

Inbetriebnahme Microsoft Graph

Microsoft Graph

Inhaltsverzeichnis

1 Voraussetzungen	3
2 Installation Microsoft Graph SDK für Powershell	3
3 Anwendung in Microsoft Entra Admin Center registrieren	3
4 Authentifizierung	4
5 Selbstsigniertes Zertifikat erstellen	5
6 Zertifikat hinterlegen	5
7 API-Berechtigungen für Anwendung konfigurieren	7
8 Test der Microsoft Graph SDK in Powershell	9

1 Voraussetzungen

Damit das Microsoft Graph Powershell SDK erfolgreich installiert werden kann, sind zunächst einige technische Voraussetzungen zu erfüllen (vgl. <https://learn.microsoft.com/en-us/powershell/microsoftgraph/installation?view=graph-powershell-1.0>)

- Upgrade auf PowerShell 5.1 oder höher
- Installieren von .NET Framework 4.7.2 oder höher
- Aktualisieren Sie PowerShellGet mit *Install-Module PowerShellGet* auf die neueste Version
- Die Richtlinie für die Ausführung von PowerShell-Skripten muss auf remote signiert oder weniger restriktiv eingestellt sein ggfs. *Set-ExecutionPolicy - ExecutionPolicy RemoteSigned* ausführen.

2 Installation Microsoft Graph SDK für Powershell

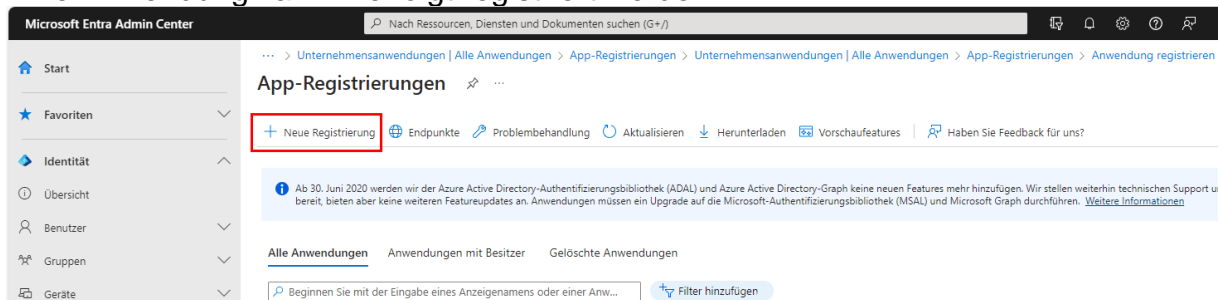
Auf den Dispatcher Servern, die für den Zugriff auf die Microsoft Azure Cloud genutzt werden sollen, muss das Microsoft.Graph Modul für die Powershell installiert werden:

Install-Module Microsoft.Graph -Scope AllUsers

3 Anwendung in Microsoft Entra Admin Center registrieren

Für die Microsoft Graph SDK für Powershell wird eine Anwendungs-Authentifizierung benötigt. Dafür muss in dem Microsoft Entra Admin Center eine Anwendung angelegt und entsprechend konfiguriert werden.

Eine Anwendung kann wie folgt registriert werden:



Microsoft Entra Admin Center

Nach Ressourcen, Diensten und Dokumenten suchen (G+/)

Start

Favoriten

Identität

Übersicht

Benutzer

Gruppen

Geräte

Anwendungen

Unternehmensanwendungen

App-Registrierungen

Schutz

Identity Governance

Azure AD External Identities

Mehr anzeigen

Schutz

Learn und Support

App-Registrierungen > Anwendung registrieren > App-Registrierungen > Unternehmensanwendungen | Alle Anwendungen > App-Registrierungen >

Anwendung registrieren

* Name

Der dem Benutzer gezeigte Anzeigename für diese Anwendung. (Dieser kann später geändert werden.)

Unicat Operations Manager Graph

Unterstützte Kontotypen

Wer kann diese Anwendung verwenden oder auf diese API zugreifen?

☒ Nur Konten in diesem Organisationsverzeichnis (nur "Unicat unified computing and technology GmbH" – einzelner Mandant)

☐ Konten in einem beliebigen Organisationsverzeichnis (beliebiger Microsoft Entra ID-Mandant – mandantenfähig)

☐ Konten in einem beliebigen Organisationsverzeichnis (beliebiger Microsoft Entra ID-Mandant – mandantenfähig) und persönliche Microsoft-Konten (z. B. Skype, Xbox)

☐ Nur persönliche Microsoft-Konten

[Entscheidungshilfe...](#)

Umleitungs-URI (optional)

Die Authentifizierungsantwort wird nach erfolgreicher Authentifizierung des Benutzers an diesen URI zurückgegeben. Die Angabe ist zum jetzigen Zeitpunkt optional und kann später geändert werden. Für die meisten Authentifizierungsszenarien ist jedoch ein Wert erforderlich.

Plattform auswählen Beispiel: https://example.com/auth

Registrieren Sie eine App, an der Sie gerade arbeiten. Integrieren Sie Katalog-Apps und andere Apps von außerhalb Ihrer Organisation, indem Sie sie aus [Unternehmensanwendungen](#)

Indem Sie den Vorgang fortsetzen, stimmen Sie den [Microsoft-Plattformrichtlinien](#) zu.

Registrieren

Microsoft Entra Admin Center

Nach Ressourcen, Die

Home > Unicat unified computing and technology GmbH > Unternehmensanwendungen | Alle Anwendungen >

Unicat Operations Manager Graph

Suche

Löschen Endpunkte Vorschaufeatures

Übersicht

Schnellstart

Integrations-Assistent

Verwalten

Branding und Eigenschaften

Authentifizierung

Zusammenfassung

Anzeigename : [Unicat Operations Manager Graph](#)

Anwendungs-ID (Client) : [89f157b-e8b7-4072-a08f-5171b0c0b0d](#)

Objekt-ID : [f53d974-c1e7-4c0f-a1b2-bef571a761eb](#)

Verzeichnis-ID (Mandant) : [600b6c7b-1c71-4812-b0a1-4307a27e7017](#)

Unterstützte Kontotypen : [Nur meine Organisation](#)

Nachdem die Anwendung erfolgreich in der Cloud registriert wurde, muss die Anwendungs-ID in das Attribut GraphClientId und die Verzeichnis-ID in das Attribut GraphTenantId an dem MSOLService-Konfigurationsobjekt in der HDB hinterlegt werden.

4 Aufentifizierung

Als nächstes muss die Anwendungs-Authentifizierung eingerichtet werden. Hierfür wird ein privates Zertifikat mit Passwort für die Dispatcher Server und das damit korrespondierende öffentliche Zertifikat für die Anwendung in der Microsoft Cloud benötigt.

Unsere Empfehlung:

Lassen Sie ein Zertifikat von einer vertrauenswürdigen dritten Zertifizierungsstelle erstellen, welches für die Authentifizierung an der Microsoft Cloud verwendet wird.

Alternativ kann auch ein selbstsigniertes Zertifikat benutzt werden.

ACHTUNG:

Selbstsignierte Zertifikate sind digitale Zertifikate, die nicht von einer vertrauenswürdigen dritten Zertifizierungsstelle signiert sind. Selbstsignierte Zertifikate werden von dem Unternehmen oder Entwickler erstellt, ausgestellt und signiert, der für die zu signierende Website oder Software verantwortlich ist. Aus diesem Grund gelten selbstsignierte Zertifikate für öffentlich zugängliche Websites und Anwendungen als unsicher.

Ist ein entsprechendes Zertifikat vorhanden, kann das nachfolgende Kapitel übersprungen werden.

5 Selbstsigniertes Zertifikat erstellen

Beispiel Zertifikat über Powershell erstellen

```
$certname = "{certificateName}" ## {certificateName} durch Zertifikatsnamen  
ersetzen  
$cert = New-SelfSignedCertificate -Subject "CN=$certname" -CertStoreLocation  
"Cert:\CurrentUser\My" -KeyExportPolicy Exportable -KeySpec Signature -KeyLength  
2048 -KeyAlgorithm RSA -HashAlgorithm SHA256
```

Zertifikat exportieren

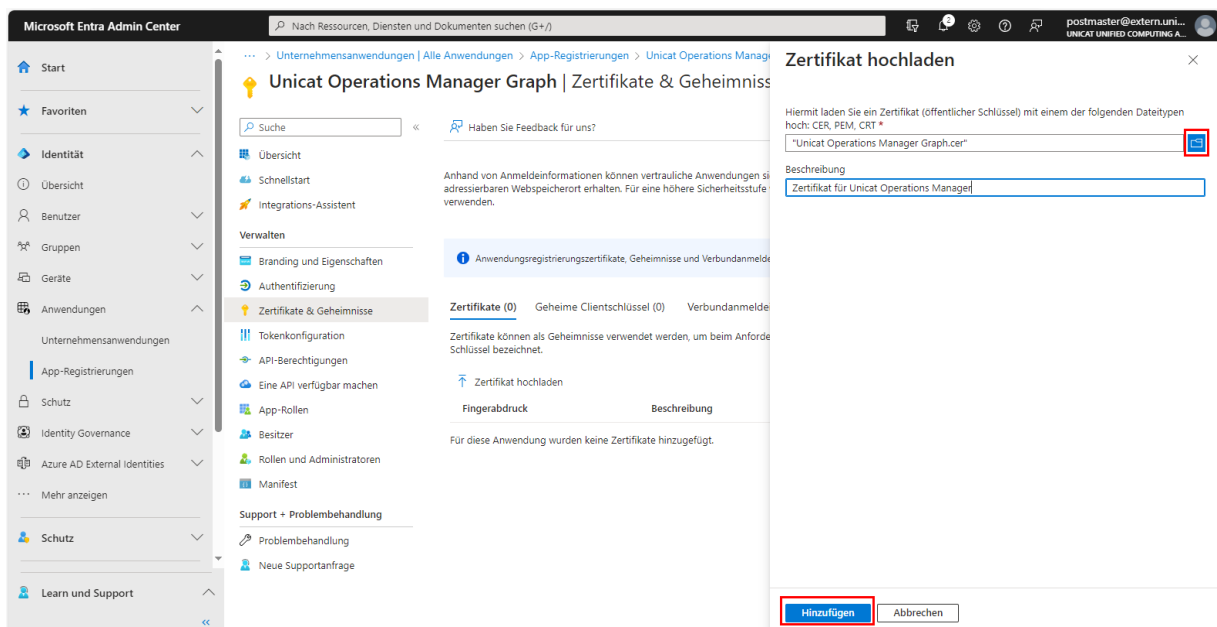
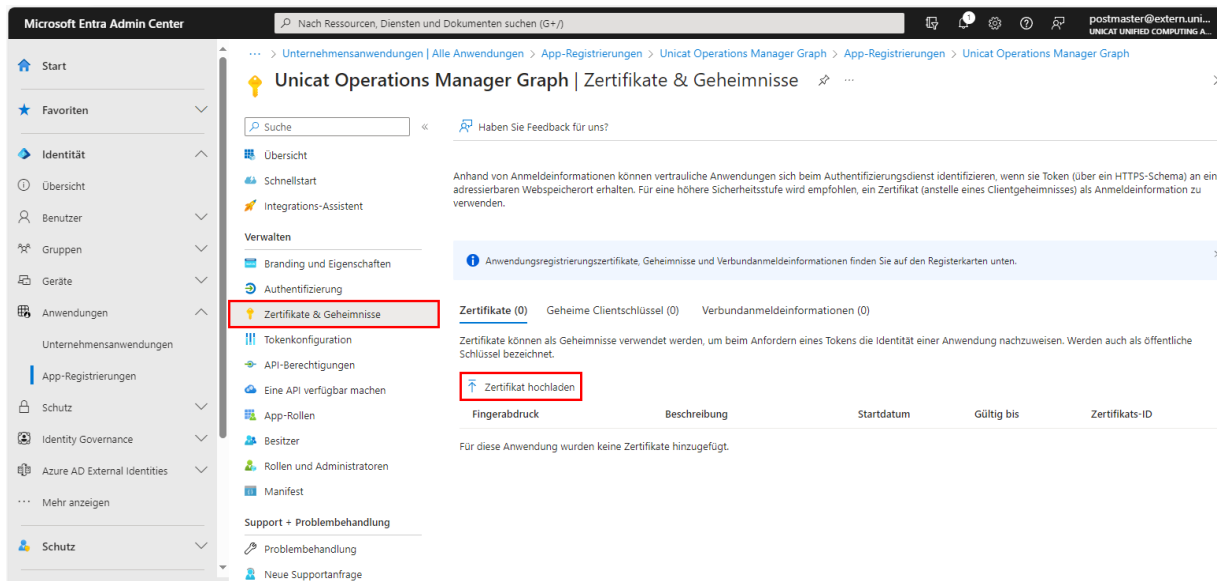
```
Export-Certificate -Cert $cert -FilePath "{exportPath}\$certname.cer" ##  
{exportPath} durch Pfad ersetzen
```

Privates Zertifikat mit Passwort exportieren

```
$mypwd = ConvertTo-SecureString -String "{myPassword}" -Force -AsPlainText ##  
{myPassword} durch Passwort ersetzen  
Export-PfxCertificate -Cert $cert -FilePath "{exportPath}\$certname.pfx" -Password  
$mypwd ## {exportPath} durch Pfad ersetzen
```

6 Zertifikat hinterlegen

Nachdem das Zertifikat erfolgreich exportiert wurde, muss nun das Zertifikat (.pfx) an der Anwendung in der Microsoft Cloud hinterlegt werden:



Zusätzlich muss das Zertifikat (.pfx) auf den Dispatcher Servern in den „Local Machine“ Zertifikatsspeicher unter „Personal“ installiert werden.

WICHTIG: Damit die Elementar Operation auf das Zertifikat zugreifen kann, muss der Service-Account welcher als Anmeldeinformation an dem Host-Objekt in der HDB hinterlegt ist, auf das Zertifikat berechtigt werden. Für die Modul-Funktion muss zusätzlich noch der lokale Benutzer „IIS_IUSRS“ berechtigt werden!

Wurden das Zertifikat entsprechend hochgeladen und installiert, muss der Name des Zertifikats an dem MSOLService-Konfigurationsobjekt in der HDB im Attribut „GraphCertificateName“ hinterlegt werden.

7 API-Berechtigungen für Anwendung konfigurieren

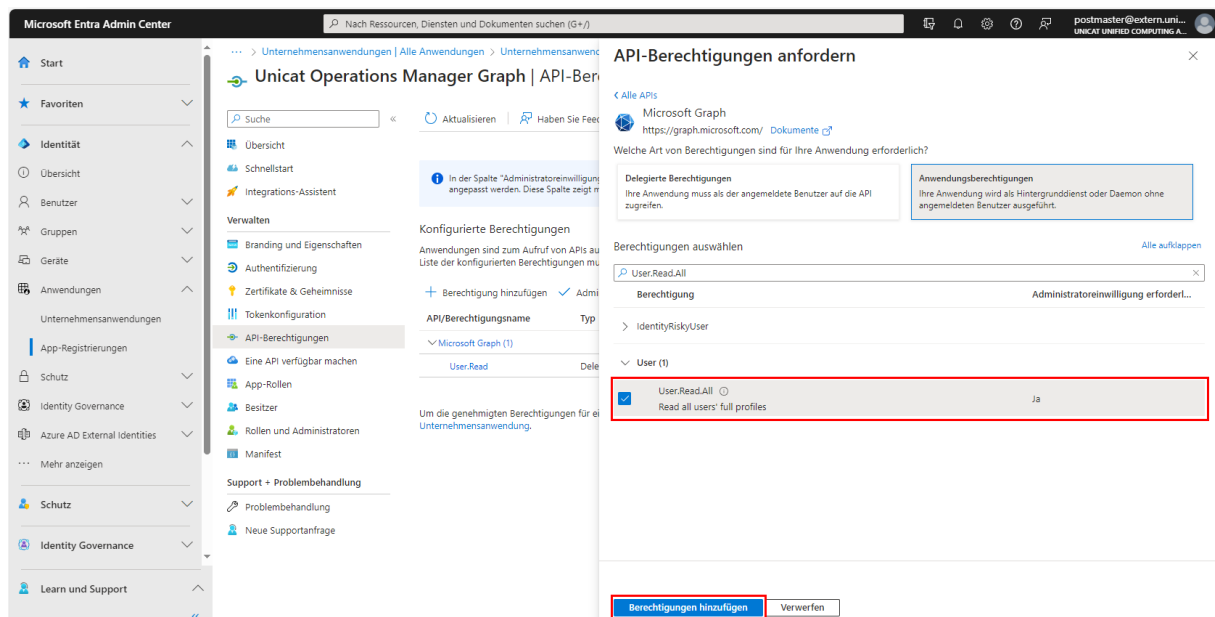
Zum Abschluss müssen noch die entsprechenden API-Berechtigungen für die Anwendung vergeben werden. Für unser Beispiel benötigen wir die Berechtigung User.Read.All. Dies kann sich aber je nach Anforderung unterscheiden!

The screenshot shows the Microsoft Entra Admin Center interface. The left sidebar contains navigation options like Start, Favoriten, Identität, and others. The main content area is titled 'Unicat Operations Manager Graph | API-Berechtigungen'. It displays a list of configured permissions under the heading 'Konfigurierte Berechtigungen'. A red box highlights the '+ Berechtigung hinzufügen' button. Below this, a table lists the permissions:

API/Berechtigungsnamen	Typ	Beschreibung	Administratoreinwill...	Status
Microsoft Graph (1)				
User.Read	Delegiert	Anmelden und Benutzerprofil lesen	Nein	...

The screenshot shows the 'API-Berechtigungen anfordern' (Request API permissions) dialog in the Microsoft Entra Admin Center. It prompts the user to select an API. Under the 'Häufig verwendete Microsoft-APIs' (Frequently used Microsoft APIs) section, a red box highlights the 'Microsoft Graph' option. Other visible options include Azure Communication Services, Azure Rights Management Services, Azure Service Management, Dynamics 365 Business Central, Dynamics CRM, Intune, Office 365 Management APIs, OneNote, and Power Automate.

The screenshot shows the 'API-Berechtigungen anfordern' (Request API permissions) dialog in the Microsoft Entra Admin Center. It prompts the user to select an API. Under the 'Welche Art von Berechtigungen sind für Ihre Anwendung erforderlich?' (Which type of permissions are required for your application?) section, a red box highlights the 'Delegierte Berechtigungen' (Delegated permissions) option. Other visible options include 'Anwendungsberechtigungen' (Application permissions).



Für den Einsatz der Sync- und Discover Funktionen des NSAzure Modul sind die folgenden Berechtigungen einzurichten:

[Home](#) > [App registrations](#) > [DotNet Graph Cert Test](#)

[DotNet Graph Cert Test | API permissions](#)

Search Refresh Got feedback?

Overview
Quickstart
Integration assistant
Diagnose and solve problems

Manage

Branding & properties
Authentication
Certificates & secrets
Token configuration
API permissions
Expose an API
App roles
Owners
Roles and administrators
Manifest

Support + Troubleshooting

New support request

Successfully granted admin consent for the requested permissions.

or in organizations where this app will be used. [Learn more](#)

Configured permissions

Applications are authorized to call APIs when they are granted permissions by users/admins as part of the consent process. The list of configured permissions should include all the permissions the application needs. [Learn more about permissions and consent](#)

+ Add a permission ✓ Grant admin consent for Unicat unified computing and technology GmbH

API / Permissions name	Type	Description	Admin consent req...	Status
Microsoft Graph (15)				
Directory.Read.All	Application	Read directory data	Yes	✓ Granted for Unicat unifi...
Directory.ReadWrite.All	Application	Read and write directory data	Yes	✓ Granted for Unicat unifi...
Domain.Read.All	Application	Read domains	Yes	✓ Granted for Unicat unifi...
Domain.ReadWrite.All	Application	Read and write domains	Yes	✓ Granted for Unicat unifi...
Group.Read.All	Application	Read all groups	Yes	✓ Granted for Unicat unifi...
MailboxFolder.Read.All	Application	Read all the users' mailbox folders	Yes	✓ Granted for Unicat unifi...
MailboxFolder.ReadWrite.All	Application	Read and write all the users' mailbox ...	Yes	✓ Granted for Unicat unifi...
MailboxItem.ImportExport.All	Application	Allows the app to perform backup an...	Yes	✓ Granted for Unicat unifi...
MailboxItem.Read.All	Application	Read all the users' mailbox items	Yes	✓ Granted for Unicat unifi...
MailboxSettings.Read	Application	Read all user mailbox settings	Yes	✓ Granted for Unicat unifi...
MailboxSettings.ReadWrite	Application	Read and write all user mailbox settin...	Yes	✓ Granted for Unicat unifi...
Policy.ReadWrite.AuthenticationFlows	Application	Read and write authentication flow p...	Yes	✓ Granted for Unicat unifi...
User.Read.All	Application	Read all users' full profiles	Yes	✓ Granted for Unicat unifi...
UserAuthenticationMethod.Read.All	Application	Read all users' authentication methods	Yes	✓ Granted for Unicat unifi...
UserAuthenticationMethod.ReadWrite.All	Application	Read and write all users' authenticati...	Yes	✓ Granted for Unicat unifi...

8 Test der Microsoft Graph SDK in Powershell

Die Microsoft Graph SDK kann in der Powershell mit dem folgenden Skript getestet werden:

```
## Verbindung zu Microsoft Graph aufbauen  
Connect-MgGraph -ClientId "{MSOLService.GraphClientID}" -TenantId  
"{MSOLService.GraphTenantId}" -CertificateName  
"CN={MSOLService.GraphCertificateName}"
```

```
## Ausgabe der Cloud User  
Get-MgUser
```

```
## Verbindung zu Microsoft Graph trennen  
Disconnect-Graph
```